

# Capsule Reviews

---

**The Capsule Reviews are intended to provide a short succinct review of each paper in the issue in order to bring the content to a wider readership. This issue's Capsule Reviews were compiled by Fairouz Kamareddine. Professor Kamareddine is an Associate Editor of *The Computer Journal* and is based in the School of Mathematical and Computer Sciences at Heriot-Watt University, Edinburgh, UK.**

---

## **An Architecture for Kernel-Level Verification of Executables at Run Time.** L. CATUOGNO AND I. VISCONTI

External intrusion is a serious threat to the security of a system connected to a network. There are two forms of external intrusions: weak and strong. Both forms involve the attacker exploiting a bug of a network demon to obtain superior privileges. In the weak form of intrusion, the attacker's control of the system can be detected by the administrator of the system and terminated by rebooting the system for instance, forcing the attacker to try to find new ways of attacking the system. In the strong form of intrusion, the attacker tries to maintain control over the system even across re-boots. This paper presents a means of reducing a strong form of intrusion into a weak one. This is done by presenting a security architecture that prevents installation of malicious code that can be executed across re-boots. The architecture can be used for any type of executable file format, provided that a tool for the signature and a kernel handler are available. This proposed architecture has also been implemented. The techniques used by intruders, as well as some of the proposed architectures for protecting servers, are reviewed. The paper also proposes solutions for a possible attack on the proposed architecture and discusses where the proposed approach might fail; this happens when the executed code inside a shared object is dynamically located or when an interpreter executes code that is handled by the kernel.

## **Comparing the Optimal Performance of Parallel Architectures.** K. KLONOWSKA, L. LUNDBERG, H. LENNERSTAD AND M. BROBERG

When implementing computer systems with multi-processors, one needs to decide whether run-time reallocation is to be allowed. Allowing run-time process reallocation in multi-processors with a very large number of processors is very difficult and costly. On the other hand, not allowing run-time reallocation may lead to unbalanced loads since some processors may be very busy while others are idle. The deciding factor in whether to allow run-time reallocation is performance gain. Performance gain depends on the quality of the scheduling and allocation policies and on the parallel framework under consideration. This leads to the question of how much one can gain at most by allowing run-time reallocation and using optimal scheduling and allocation. This paper answers this question for a very wide range of multi-processors and parallel programs. This is done by presenting transformations that allow one to separate execution and synchronization. Given the parameters  $n$ ,  $k$ ,  $q$ ,  $t$ ,  $z$ , the paper characterizes a set of worst-case programs

called complete programs. A formula,  $H(n, k, q, t, z)$ , is given that represents the bound on the gain of using a system with  $q$  processors and run-time process allocation compared with using a system with  $k$  processors, no reallocation and a communication delay  $t$ , for a program with  $n$  processes and a synchronization granularity  $z$ . The validations of the result lead to the conclusion that the bound is valid for all programs and that this bound is indeed optimal when  $t$  is large or  $z$  is small. The main contribution of this paper is that it handles and validates a more realistic computer model where the communication delay,  $t$ , and the granularity,  $z$ , are taken into account.

## **Video-on-Demand Based on Delayed-Multicast: Algorithmic Support.** N. GLINOS, D. B. HOANG, C. NGUYEN AND A. SYMVONIS

Building a cost-effective and scalable continuous multimedia system remains a challenge. For example, in order to build a functional video-on-demand system, one needs to overcome many technical problems including the large amount of bandwidth required both from I/O and the network. To address the network bandwidth problem, a delayed-multicast technique was developed in an earlier paper. The delayed-multicast technique increases the level of concurrency in the system, minimizes network traffic and allows us to service requests with different starting times by using a buffer. This means that instead of the server sending many streams (one per request), the delayed-multicast technique allows the bandwidth requirement from the video server to the intermediate router to be reduced to just one stream. This works for both requests whose start times are known and requests made on-line. Here, most of the algorithmic problems are related to trade-offs between bandwidth and memory. The paper is concerned with these algorithmic issues. In order to do so, the paper introduces the minimum total memory (MTM), the minimum total traffic (MTT) and the minimum maximum memory per node (MMMN) delayed-multicast allocation problems. These problems are examined on the chandelier and broom networks. Polynomial time algorithms for solving the MTM and MTT problems on the broom network are given, as well as a heuristic method for obtaining a solution for the MTM problem on tree networks. Furthermore, the paper shows that a version of the MMMN allocation problem is NP-complete on a two-level broom network.

## **Cache-Aware Real-Time Disk Scheduling.** H.-P. CHANG, R.-I. CHANG, W.-K. SHIH AND R.-C. CHANG

Delivery of data is usually constrained by real time. Nonetheless, requested data must be retrieved and delivered

on time. This paper addresses real-time data retrieval by taking into account an increased capacity of on-disk cache, which can service data requests without incurring physical disk access. As a matter of fact, if access to a disk is serviced by the on-disk cache, the I/O delay will be significantly reduced. Moreover, if the parameters of the on-disk cache are disclosed, they can be used in disk scheduling, which will help in spatial and temporal locality. However, since cache design gives a competitive edge in the market, it is either patented or considered to be a secret. This means that since the parameters of the on-disk cache may be unknown, the on-cache influence is ignored, and during the schedulability testing, worst-case assumptions of physical disk access are made, which results in a decrease of system performance. This paper presents three different cache-aware algorithms: the deadline shift scheme, the collaboration scheme and the Cache-Aware Real-Time Disk Scheduling (CARDS) scheme. In the deadline scheme, disk tasks whose access has a spatial locality are brought closer to meet their temporal locality by the deadline shift operation increasing the cache hit probability. In the collaboration scheme, tasks that have the chance to be cache hits are grouped together into a reschedulable group, and this requires a feasibility check to make sure the schedule remains feasible after re-ordering. Techniques for accelerating the checking operation are proposed in the paper. The CARDS scheme re-orders the tasks if a cache hit is guaranteed after such a re-ordering. These proposed algorithms do not depend on any specific on-disk cache. Experiments carried out using these three algorithms show that they do indeed obtain more data.

**Optimizing Object-Based Multimedia Delivery Across Bandwidth Constrained Networks.** C.-M. CHEN, H.-M. SUN AND L. C. SHU

Different users of a multimedia service may have different on-line requests. In this case, the scheduling problem with quality of service (QoS) requirements becomes important, especially if bandwidth is constrained. When the bandwidth is fixed and the multimedia streams are known *a priori*, off-line scheduling becomes appropriate and some QoS degradation (e.g. dropping or delaying unimportant multimedia data) can be tolerated. Earlier studies show that hybrid models with data dropping and transmission delays of data work better than models with only data dropping or transmission delays. This paper proposes an object-based multimedia model for describing QoS requirements such as the maximum data-dropping rate or the maximum data-delay rate. The paper optimizes the off-line scheduling of non-periodic multimedia requests when the bandwidth is limited. To measure the user satisfaction with the specified QoS requirements, the paper proposes a net-profit resource allocation model and defines rewards and penalties accordingly. The net profit is taken to be the total rewards minus the total penalties. The scheduling objective of the paper is to maximize user satisfaction with the specified QoS requirements using this net-profit model. Optimal solutions for scheduling problems are found with reasonable time complexity.

**Legality of XML-Schema Type Hierarchies.** A. FORMICA  
The eXtensible Markup Language (XML) is nowadays considered to be one of the most important standards for exchanging data across the Internet. The XML-schema has also been playing an important role in designing structured XML data. Although a lot of research has been conducted on XML-related methodologies and applications (and in different areas of computer science), much remains to be done on this rather new and influential XML phenomenon. In particular, formal methodologies addressing the design of XML-schema type hierarchies have not been proposed. This paper addresses this problem and proposes a formal framework supporting the design of XML-schema type hierarchies, together with a definition of a legal type hierarchy and an algorithm for checking the legality of an XML-schema type hierarchy. Although the proposed framework does not deal with constructors like union and choice, the paper is a good step in the right direction.

**Optimal Algorithms for  $2 \times n$  Mastermind Games—a Graph-Partition Approach.** S.-T. CHEN AND S.-S. LIN

The game of Mastermind is a deductive game for two players: a codemaker and a codebreaker. The codemaker chooses a secret code of four pegs  $(s_1, s_2, s_3, s_4)$ , each chosen from one of six colours. The codebreaker tries to guess the code. After each guess  $(g_1, g_2, g_3, g_4)$ , the codemaker responds with a pair of numbers  $[B, W]$ , where  $B$  is the number of positions,  $j$ , such that  $s_j = g_j$  and  $W$  is the number of positions,  $j$ , such that  $s_j \neq g_j$  but  $s_j = g_k$  for some  $k$  different from  $j$ . The codebreaker tries to guess the code based on these responses. Different strategies for solving this problem (where four pegs and six colours are used) have been proposed. However, the complexity of the problem grows exponentially when the number of pegs or the number of colours is taken to be arbitrary (rather than four or six, respectively). This paper develops a methodology to find the optimal strategies for general Mastermind games with two pegs and  $n$  colours (for  $n \geq 2$ ). The paper exploits properties of game trees to calculate the number of guesses required in the worst and expected cases for a deductive game. Moreover, the paper shows that both worst and expected cases can be solved with the same strategy. Furthermore, the game-guessing process is represented on a graph model that helps identify some recursive and symmetric structures, reduces the search space and helps derive optimal strategies more efficiently.

**Loopless Array Generation of Multiset Permutations.** J. F. KORSH AND P. S. LAFOLLETTE

An algorithm for generating combinatorial objects is said to be loopless if a fixed number of constant time instructions is used to generate an object from its predecessor. Loopless algorithms to generate all the permutations of a multiset have already been given. Such known algorithms that use linear storage require a linked list to represent the permutations. This paper presents a loopless algorithm to generate all the permutations of a multiset using an array for the permutations and still only requiring linear storage.